Nonmonotonic Tools for Argumentation

Gerhard Brewka

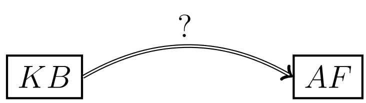
Universität Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany
brewka@informatik.uni-leipzig.de

Abstract. Dung's argumentation frameworks (AFs) have become very popular as semantical tools in argumentation. We discuss a generalization of AFs called abstract dialectical frameworks (ADFs). These frameworks are more flexible in that they allow arbitrary boolean functions to be used for the specification of acceptance conditions for nodes. We present the basic underlying definitions and give an example illustrating why they are useful. More precisely, we show how they can be used to provide a semantical foundation for Gordon, Prakken and Walton's Carneades model of argumentation, lifting the limitation of this model to acyclic argument graphs.

1 Introduction

Dung's abstract argumentation frameworks (AFs) [4] are without doubt the most influential tools currently used in argumentation. They provide basic conflict handling mechanisms for argumentation: whenever all relevant arguments and the conflict relations among them have been established, the semantics defined for AFs specify different ways of identifying reasonable subsets of the arguments which are jointly acceptable.

AFs have become extremely popular in argumentation. They are commonly used in the following way: given a knowledge base, say consisting of defeasible rules, preferences, proof standards, etc., the available information is first compiled into adequate arguments and attacks. The resulting AF then provides the system with a choice of different semantics. The following picture illustrates this:



AFs are among the simplest nonmonotonic systems one can think of - and this is certainly part of why they are so popular. Still, we believe - and will demonstrate here - that adding further functionality to AFs may be worthwhile as this can bring the target systems of the compilation described above closer to what one typically finds in the original knowledge bases.

This was one of the reasons for Brewka and Woltran to introduce abstract dialectical frameworks (ADFs) [3]. ADFs are a powerful generalization of Dung-style argumentation frameworks. Dung argumentation frameworks have an implicit, fixed criterion for the acceptance of a node in the argument graph: a

T. Janhunen and I. Niemelä (Eds.): JELIA 2010, LNAI 6341, pp. 1–6, 2010.

node is accepted iff all its parents are defeated. This acceptance criterion can be viewed as an implicit boolean function assigning a status to an argument based on the status of its parents. The basic idea underlying ADFs is to make this boolean function explicit, and then to allow arbitrary acceptance conditions for nodes to be specified. In a nutshell, this turns the "calculus of opposition" provided by AFs into a "calculus of support an opposition".

It turns out that the standard semantics for Dung frameworks - grounded, preferred and stable - can be generalized to ADFs, the latter two to a slightly restricted class of ADFs called bipolar, where each link in the graph either supports or attacks its target node. Since all ADFs we are dealing with here are bipolar, we will simply speak of ADFs and omit the adjective "bipolar" whenever there is no risk of confusion.

In the next section we will briefly introduce the main definitions underlying ADFs. We will then illustrate why they are useful, showing how Carneades argument evaluation structures [6,7] can be reconstructed as ADFs.

2 Abstract Dialectical Frameworks

An ADF [3] is a directed graph whose nodes represent arguments or statements which can be accepted or not. The links represent dependencies: the status of a node s only depends on the status of its parents (denoted $par(s)$), that is, the nodes with a direct link to s. In addition, each node s has an associated acceptance condition C_s specifying the conditions under which s is accepted. This is where ADFs go beyond Dung argumentation frameworks. C_s is a boolean function yielding for each assignment of values to $par(s)$ one of the values *in*, *out* for s. As usual, we will identify value assignments with the sets of nodes which are *in*. Thus, if for some $R \subseteq par(s)$ we have $C_s(R) = in$, then s will be accepted provided the nodes in R are accepted and those in $par(s) \setminus R$ are not accepted.

Definition 1. *An* abstract dialectical framework *is a tuple $D = (S, L, C)$ where*

- *S is a set of statements,*
- *$L \subseteq S \times S$ is a set of links,*
- *$C = \{C_s\}_{s \in S}$ is a set of total functions $C_s : 2^{par(s)} \to \{in, out\}$, one for each statement s. C_s is called acceptance condition of s.*

S and L obviously form a graph, and we sometimes refer to elements of S as nodes. For the purposes of this paper we will only deal with a subset of ADFs, called bipolar in [3]. In such ADFs each link is either attacking or supporting:

Definition 2. *Let $D = (S, L, C)$ be an ADF. A link $(r, s) \in L$ is*

1. supporting *iff, for no $R \subseteq par(s)$, $C_s(R) = in$ and $C_s(R \cup \{r\}) = out$,*
2. attacking *iff, for no $R \subseteq par(s)$, $C_s(R) = out$ and $C_s(R \cup \{r\}) = in$.*

For simplicity we will only speak of ADFs here, keeping in mind that all ADFs in this paper are indeed bipolar.

It turns out that Dung's standard semantics - grounded, stable, preferred - can be generalized adequately to ADFs. We first introduce the notion of a model. Intuitively, in a model all acceptance conditions are satisfied.

Definition 3. *Let $D = (S, L, C)$ be an ADF. $M \subseteq S$ is a* model *of D if for all $s \in S$ we have $s \in M$ iff $C_s(M \cap par(s)) = in$.*

We first define the generalization of grounded semantics:

Definition 4. *Let $D = (S, L, C)$ be an ADF. Consider the operator*

$$\Gamma_D(A, R) = (acc(A, R), reb(A, R))$$

where

$$acc(A, R) = \{r \in S \mid A \subseteq S' \subseteq (S \setminus R) \Rightarrow C_r(S' \cap par(r)) = in\}, \text{ and}$$
$$reb(A, R) = \{r \in S \mid A \subseteq S' \subseteq (S \setminus R) \Rightarrow C_r(S' \cap par(r)) = out\}.$$

Γ_D is monotonic in both arguments and thus has a least fixpoint. E is the well-founded model *of D iff for some $E' \subseteq S$, (E, E') is the least fixpoint of Γ_D.*

For stable models we apply a construction similar to the Gelfond/Lifschitz reduct for logic programs. The purpose of the reduction is to eliminate models in which nodes are *in* just because of self supporting cycles:

Definition 5. *Let $D = (S, L, C)$ be an ADF. A model M of D is a* stable model *if M is the least model of the reduced ADF D^M obtained from D by*

1. *eliminating all nodes not contained in M together with all links in which any of these nodes appear,*
2. *eliminating all attacking links,*
3. *restricting the acceptance conditions C_s for each remaining node s to the remaining parents of s.*

Preferred extensions in Dung's approach are maximal admissible sets, where an admissible set is conflict-free and defends itself against attackers. This can be rephrased as follows: E is *admissible* in a Dung argumentation framework $A = (AR, att)$ iff for some $R \subseteq AR$

- R does not attack E, and
- E is a stable extension of $(AR\text{-}R, att \cap (AR\text{-}R \times AR\text{-}R))$.

This leads to the following generalization:

Definition 6. *Let $D = (S, L, C)$, $R \subseteq S$. D-R is the ADF obtained from D by*

1. *deleting all nodes in R together with their acceptance conditions and links they are contained in.*

2. *restricting acceptance conditions of the remaining nodes to the remaining parents.*

Definition 7. *Let $D = (S, L, C)$ be an ADF. $M \subseteq S$ is* admissible *in D iff there is $R \subseteq S$ such that*

1. *no element in R attacks an element in M, and*
2. *M is a stable model of D-R.*

M is a preferred *model of D iff M is (subset) maximal among the sets admissible in D.*

Brewka and Woltran also introduced weighted ADFs where an additional weight function w assigns qualitative or numerical weights to the links in the graph. This allows acceptance conditions to be defined in a domain independent way, based on the weights of links rather than on the involved statements. They also showed how the proof standards proposed by Farley and Freeman [5] can be formalized based on this idea.

The reader is referred to [3] for further details.

3 Application: Reconstructing Carneades

The Carneades model of argumentation, introduced by Gordon, Prakken and Walton in [6] and developed further in a series of subsequent papers [7,1,8], is an advanced general framework for argumentation.[1] It captures both static aspects, related to the evaluation of arguments in a particular context based on proof standards for statements and on weights arguments are given by an audience, and dynamic aspects, covering for instance the shift of proof burdens in different stages of the argumentation process.

Unlike many other approaches, Carneades does not rely on Dung's argumentation frameworks (AFs) [4] for the definition of its semantics, more specifically its notion of acceptable statements. One goal of our reconstruction is to provide a link, albeit an indirect one, between Carneades and AFs. As we will see, both are instances of a more general framework. Moreover, in spite of this generality, Carneades suffers from a restriction: it is assumed that the graphs formed by arguments are acyclic. This is not as bad as it may first sound, as the use of pro and con arguments allows some conflicts to be represented which require cyclic representations in other frameworks. Still, cycles in argumentation appear so common that forbidding them right from the start is certainly somewhat problematic. And indeed, the authors in [6] write (page 882):

> *"We ... leave an extension to graphs that allow for cycles through exceptions for future work."*

[1] As of June 2010, [6] is among the 10 most cited papers which appeared in the Artificial Intelligence Journal over the last 5 years.

Indeed, by reconstructing Carneades argument evaluation structures as ADFs, the mentioned limitation can be overcome.

We cannot go into the technical details of the translation here and refer the reader to [2]. Nevertheless, we want to give a basic idea how the translation works. We start with the definition of arguments in Carneades [7]:

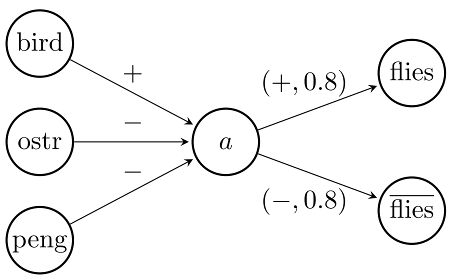
Definition 8 (argument). *Let $\mathcal{L}$ be a propositional language. An* ***argument*** *is a tuple $\langle P, E, c\rangle$ where $P \subset \mathcal{L}$ are its* ***premises****, $E \subset \mathcal{L}$ with $P \cap E = \emptyset$ are its* ***exceptions*** *and $c \in \mathcal{L}$ is its* ***conclusion****. For simplicity, c and all members of P and E must be literals, i.e. either an atomic proposition or a negated atomic proposition. Let p be a literal. If p is c, then the argument is an argument* ***pro*** *p. If p is the complement of c, then the argument is an argument* ***con*** *p.*

An argument evaluation structure was defined in [7] as a triple consisting of a stage, an audience, and a function assigning a proof standard to propositions. Since we are only interested in stage specific argument evaluation, the status part of the definition of stages (see [7]) can be skipped, keeping only the set of arguments, together with the audience (a pair consisting of a set of assumptions and a weight function) and the proof standards.

To illustrate our translation, consider the argument

$$a = \langle \{bird\}, \{peng, ostr\}, flies\rangle$$

and assume $weights(a) = 0.8$. The ADF graph generated by this argument is shown in the following figure (we mark links with their weights):



Proof standards and assumptions can be captured by adequate acceptance conditions for the nodes in the ADF. It was proven in [2] that this translation yields the desired results, that is, the acceptable statements in Carneades coincide with the statement nodes assigned *in* in the generated ADF.

The real advantage of our translation is that we can now lift the restriction of acyclicity. Nothing in the translation hinges on the fact that the set of Carneades arguments is acyclic. Indeed, cycles in the set of Carneades arguments will lead to cycles in the ADF, yet these cycles are handled - in different ways - by the available semantics of ADFs.

These results are of interest, both from the point of view of ADFs and from the point of view of Carneades:

1. They show that ADFs not only generalize Dung argumentation frameworks - which have been the starting point for their development. They also generalize Carneades argument evaluation structures.
2. They clarify the relationship between Carneades and Dung AFs, showing that both are instances of ADFs. They thus help to put Carneades on an equally solid formal foundation.
3. Finally, they allow us to lift the restriction of Carneades to acyclic argument structures.

As we believe, this provides sufficient evidence that the ADF framework is indeed a useful nonmonotonic tool in the theory of argumentation.

References

1. Ballnat, S., Gordon, T. : Goal selection in argumentation processes. In: Proc. Computational Models of Argumentation (2010)
2. Brewka, G., Gordon, T.F.: Carneades and abstract dialectical frameworks: A reconstruction. In: Proc. COMMA (2010)
3. Brewka, G., Woltran, S.: Abstract dialectical frameworks. In: Proc. Principles of Knowledge Representation and Reasoning, pp. 102–111 (2010)
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
5. Farley, A.M., Freeman, K.: Burden of proof in legal argumentation. In: Proc. ICAIL 1995, pp. 156–164 (1995)
6. Gordon, T.F., Prakken, H., Walton, D.: The Carneades model of argument and burden of proof. Artif. Intell. 171(10-15), 875–896 (2007)
7. Gordon, T.F., Walton, D.: Proof burdens and standards. In: Rahwan, I., Simari, G. (eds.) Argumentation in Artificial Intelligence, pp. 239–258 (2009)
8. Grabmair, M., Gordon, T., Walton, D.: Probabilistic semantics for the Carneades argument model using Bayesian belief networks. In: Proc. Computational Models of Argumentation (2010)